



33^{ος}

ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑ Α΄ ΦΑΣΗΣ

ΤΡΑΠΕΖΙΚΕΣ ΣΥΝΑΛΛΑΓΕΣ

Η συνεταιριστική τράπεζα της γειτονιάς σας ζητάει τη βοήθειά σας στην υλοποίηση του μηχανογραφικού της συστήματος. Οι κωδικοί των τραπεζικών λογαριασμών είναι το πολύ εννιαψήφιοι θετικοί ακέραιοι αριθμοί. Οι πελάτες εκτελούν τριών ειδών συναλλαγές:

- κατάθεση στο λογαριασμό A ποσού X ευρώ
- ανάληψη από το λογαριασμό A ποσού X ευρώ
- ερώτηση υπολοίπου του λογαριασμού A

Αρχικά θεωρείται ότι όλοι οι λογαριασμοί έχουν μηδενικό υπόλοιπο. Για να εκτελεστεί μία ανάληψη πρέπει να υπάρχει διαθέσιμο το ζητούμενο ποσό στο λογαριασμό, διαφορετικά εκτυπώνεται μήνυμα σφάλματος.

Πρόβλημα

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI (PASCAL, C, C++, Java) το οποίο θα διαβάζει μία ακολουθία από συναλλαγές και θα τις εκτελεί μία-μία, με τη σειρά που δίνονται.

Αρχεία εισόδου:

Τα αρχεία εισόδου με όνομα **bankacc.in** είναι αρχεία κειμένου με την εξής δομή: Στην πρώτη γραμμή υπάρχει ένας ακέραιος αριθμός **N**, το πλήθος των συναλλαγών που ακολουθούν. Κάθε μία από τις επόμενες **N** γραμμές παριστάνει μία συναλλαγή και αρχίζει με ένα από τα γράμματα "d" (κατάθεση, deposit), "w" (ανάληψη, withdrawal) ή "q" (ερώτηση, query). Ακολουθούν, χωρισμένα με ένα κενό διάστημα:

- στην περίπτωση κατάθεσης ή ανάληψης ("d" ή "w") δύο ακέραιοι αριθμοί **A** και **X**: ο αριθμός λογαριασμού και το ποσό
- στην περίπτωση ερώτησης υπολοίπου ("q") ένας ακέραιος αριθμός **A**: ο αριθμός λογαριασμού



Αρχεία εξόδου:

Τα αρχεία εξόδου με όνομα **bankacc.out** είναι αρχεία κειμένου με την εξής δομή. Περιέχουν ακριβώς **N** γραμμές, μία για κάθε συναλλαγή που πραγματοποιείται. Η γραμμή αυτή πρέπει να περιέχει:

- στην περίπτωση κατάθεσης ή ανάληψης που επιτυγχάνει, το γράμμα "s" (επιτυχία, success)
- στην περίπτωση ανάληψης που αποτυγχάνει λόγω μη διαθέσιμου υπολοίπου, το γράμμα "f" (αποτυχία, failure)
- στην περίπτωση ερώτησης υπολοίπου, έναν ακέραιο αριθμό: το τρέχον υπόλοιπο του αντίστοιχου λογαριασμού

Περιορισμοί:

- $1 \leq N \leq 1.000.000$
- $1 \leq A \leq 999.999.999$
- $1 \leq X \leq 1.000.000.000$
- Το υπόλοιπο κάθε τραπεζικού λογαριασμού δε θα υπερβαίνει ποτέ το 1.000.000.000

Παραδείγματα αρχείων εισόδου-εξόδου:

bankacc.in	bankacc.out
8	s
d 6 1000	0
q 4	s
d 4 500	500
q 4	f
w 4 750	s
w 6 200	800
q 6	500
q 4	

Εξήγηση παραδείγματος:

Εκτελούνται κατά σειρά οι εξής οκτώ συναλλαγές:

1. κατατίθενται 1000€ στο λογαριασμό 6, με επιτυχία
2. ερώτηση υπολοίπου για το λογαριασμό 4, έχει μηδενικό υπόλοιπο
3. κατατίθενται 500€ στο λογαριασμό 4, με επιτυχία
4. ερώτηση υπολοίπου για το λογαριασμό 4, έχει 500€
5. ανάληψη 750€ από το λογαριασμό 4, αποτυγχάνει γιατί το υπόλοιπο είναι μόνο 500€
6. ανάληψη 200€ από το λογαριασμό 6, με επιτυχία
7. ερώτηση υπολοίπου για το λογαριασμό 6, έχει 800€ (1000-200)
8. ερώτηση υπολοίπου για το λογαριασμό 4, έχει 500€



Υποπροβλήματα:

- Για περιπτώσεις ελέγχου συνολικής αξίας 10%, οι αριθμοί λογαριασμών θα είναι μονοψήφιοι
- Για περιπτώσεις ελέγχου συνολικής αξίας 50%, οι αριθμοί λογαριασμών θα είναι το πολύ εξαψήφιοι

Παρατηρήσεις:

Μορφοποίηση: Στην είσοδο αλλά και στην έξοδο, κάθε γραμμή τερματίζει με έναν χαρακτήρα `newline`.

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 64 MB.

Επικεφαλίδες στον πηγαίο κώδικα: Στην αρχή του πηγαίου κώδικά σας, θα πρέπει να χρησιμοποιήσετε τις παρακάτω επικεφαλίδες.

```
(* USER: username
```

```
LANG: PASCAL
```

```
TASK: bankacc *)
```

για κώδικα σε PASCAL

```
/* USER: username
```

```
LANG: C
```

```
TASK: bankacc */
```

για κώδικα σε C

```
/* USER: username
```

```
LANG: C++
```

```
TASK: bankacc */
```

για κώδικα σε C++

```
/* USER: username
```

```
LANG: Java
```

```
TASK: bankacc */
```

για κώδικα σε Java